



Programación Tarea Semestral

Javier Vidal Valenzuela

28 de septiembre de 2022

Introducción

El propósito de esta tarea es familiarizarse con el proceso de análisis, diseño e implementación de la solución a un problema de programación de mediana a alta complejidad usando diagramas de flujo y lenguaje Python. También se espera que a través de la experiencia en este trabajo se vean fortalecidas las habilidades de trabajo en equipo.

El trabajo **debe ser realizado en grupos de 3 ó 4 alumnos/as, todos de una misma sección o curso**, quedando absolutamente prohibido el trabajo individual o en una configuración distinta de equipos.

Este enunciado se entregará a través de la plataforma CANVAS y en el caso de requerir aclaración de alguno de los aspectos expuestos en el documento, se recomienda que la comunicación se haga electrónicamente, de preferencia usando CANVAS, para hacer extensivas las explicaciones al resto del curso.

POKER DE DADOS

Para jugar al *Póker de Dados* se requiere al menos 4 jugadores, cada uno con un “cacho” con 5 dados.



Un jugador, al lanzar un cacho, puede obtener las siguientes jugadas (en orden de menor a mayor valor):

- Pareja, si tiene dos dados iguales. Si dos o más jugadores comparten la misma jugada gana la pareja con los dados de mayor valor, siendo el AS (el 1) el valor mayor.
- Doble pareja, si tiene 2 parejas. Si entre dos o más jugadores hay empate manda la pareja de mayor valor.
- Trio, si tiene tres dados iguales. Si dos o más jugadores comparten la misma jugada gana el trio con los dados de mayor valor.
- Full, si tiene una pareja y un trio. Si dos o más jugadores comparten la misma jugada gana el trio con los dados de mayor valor.
- Escala, si los dados están en secuencia. Puede ser **As, 2, 3, 4, 5** o **2, 3, 4, 5, 6**, la primera secuencia tiene mayor valor.
- Póker, Si tiene cuatro dados iguales. Si dos o más jugadores comparten la misma jugada gana el póker con los dados de mayor valor.

-
- Repóker, Si tiene los cinco dados iguales. Si dos o más jugadores comparten la misma jugada gana el repóker con los dados de mayor valor.

En esencia, el juego trata de lanzar y determinar cuál jugador obtuvo el mejor juego en uno o dos lanzamientos. La dinámica del juego sigue más o menos el siguiente procedimiento ¹:

1. Todos los jugadores ponen sus dados en sus cachos y los lanzan boca abajo.
2. Cada jugador elige los dados que representen su mejor jugada y la mencionan a sus oponentes.
3. Todos los jugadores, incluyendo el que tenga la mejor jugada, tienen derecho a elegir dados para lanzar de nuevo y mejorar su jugada anterior.
4. Nuevamente, los jugadores que hayan decidido realizar su segunda jugada, ponen sus dados en sus cachos y los lanzan boca abajo.
5. Nuevamente, cada jugador elige los dados que representen su mejor jugada y la mencionan a sus oponentes.
6. Gana la jugada de mayor valor si existe.

Caja jugada ganada valdrá un punto, el jugador que alcance la cantidad de puntos convenida al principio del juego entre los jugadores será el triunfador absoluto del juego.

Se pide implementar un simulador de juego de póker de dados en Python en que participen n jugadores (no más de 16 jugadores en total), uno será Usted. El simulador debe consultar la cantidad de puntos p a la que se disputará la partida.

En el simulador, los lanzamientos de los dados de todos los jugadores serán números entre 1 y 6 aleatorios, generados por el computador usando la biblioteca RANDOM.

Cada jugador, menos Ud., se quedará con la mejor jugada obtenida en cada tirada. En el caso de Usted, podrá elegir su jugada y los dados que usará para la segunda tirada en cada juego.

El jugador que llegue primero a los p puntos gana la partida.

Entradas:

Las entradas a este programa serán las siguientes:

- La cantidad de jugadores n con n mayor o igual que 4 y menor o igual a 16.
- La cantidad de puntos p que debe alcanzar el jugador para ganar, p debe ser un entero positivo.

Además, un jugador (Usted) podrá, en cada jugada, elegir su jugada y los dados que lanzará de nuevo.

Salidas:

La salida de este programa consistirá en lo siguiente:

Por cada jugada se debe mostrar los dados obtenidos por cada jugador, incluyendo su jugada.

Por cada jugada y jugador, menos Usted, se debe mostrar el nombre de la mejor jugada.

Por cada jugador se irá visualizando y actualizando los puntos obtenidos.

Ejemplo de entrada y salida:

Los ejemplos de entrada y salida se proporcionarán más adelante.

¹Esta descripción no es exacta a lo que es el juego real, sino que será la que se espera que se implemente en el juego

Instrucciones

Este trabajo debe ser realizado por los equipos establecidos en la sección Personas - EQUIPOS TS en CANVAS, cuyos integrantes deben designar un **coordinador/a o líder**, quien será el/la que se comunique con el profesor en caso de requerirlo. La conformación de los equipos puede ser modificada hasta el **viernes 30 de septiembre de 2022**.

Los alumnos que no estén registrados en ningún equipo serán designados en equipos conformados por el profesor el día lunes 3 de octubre de 2022.

Habrán dos entregas de la tarea. La primera entrega será el **lunes 24 de octubre** y consistirá en el **análisis del problema**. Esto es, la definición de los aspectos de la descripción del problema no incluidos en el enunciado más una descripción detallada del método matemático, geométrico, estadístico u otro que permite resolver el problema, incluyendo detalles de funcionamiento que permitan relevar la forma en que opera la situación a automatizar. En definitiva, un informe que dé cuenta de la comprensión detallada del problema y de cómo será resuelto.

La segunda entrega será la definitiva, cuyo **plazo máximo vence el lunes 28 de noviembre de 2022**, para lo cual cada equipo debe preparar lo siguiente:

1. un archivo denominado **INFORME.PDF** que contenga:
 - una **portada** en que figuren, como mínimo, los nombres de los integrantes del equipo, carrera y el nombre del curso.
 - En su interior, el informe debe establecer **qué funciones fueron asignadas a cada miembro del grupo de trabajo**, es decir, la **organización del equipo**,
 - se debe además describir el **desarrollo** indicando **la forma en que se realizó el trabajo (análisis del problema)** (descripción detallada del método matemático, geométrico, estadístico o de otra disciplina que permite resolver el problema), **estrategias de solución** (algoritmos, funciones, estructuras de datos, etc. que implementan la solución al problema) e incluya algunas **pruebas** realizadas con datos.
 - Finalmente, el informe debe incluir también algunas **conclusiones generales** del trabajo.

No incluya código en el informe.

2. un archivo denominado **readme.txt** que contenga **los nombres de los integrantes del equipo y las instrucciones**, tanto para ejecutar su programa. Las indicaciones en este archivo se seguirán estrictamente, por lo tanto, es su responsabilidad ser claro en el orden y especificidad de las instrucciones.
3. uno o más archivos ***.py** o ***.ipynb** **con el código en lenguaje Python**. Los programas deben incluir (en líneas de comentarios) los nombres completos de los integrantes del grupo, carrera y sección o curso al que pertenecen. Además deben incluir comentarios que permitan identificar las partes principales de los programas.

Todos los archivos mencionados deben comprimirse en un paquete ZIP o RAR con el nombre identificador de su equipo, por ejemplo "Equipo_001.ZIP", el cual debe ser enviado a través de la plataforma CANVAS.

- Asegurarse de escribir la información de identificación del grupo en todos los archivos.
- Eliminar despliegues innecesarios en el(los) programa(s).
- No serán recibidas partes o la totalidad de la tarea en forma impresa.
- No envíe archivos ejecutables (*.EXE), la plataforma y algunos clientes de correo electrónico no lo permiten.
- Cuando envíe su trabajo, el profesor confirmará si su tarea ha sido recibida sin problemas, si esto no ocurre, consulte al profesor.
- Si después de enviar la tarea descubre un error y el grupo desea enviar una copia revisada, hacer otro paquete ZIP o RAR y publicar en INFOALUMNO (no reemplazar el archivo anterior).

Evaluación

El código será compilado, ejecutado y probado en una máquina usando WINDOWS 10, JUPYTER NOTEBOOK, por lo tanto, si algún grupo requiere la instalación de alguna biblioteca especial o aplicación complementaria, o bien, si es necesario compilar o correr la aplicación en un sistema operativo diferente, debe informarlo al profesor usando el archivo `readme.txt`.

El puntaje máximo de la tarea es 60 puntos distribuidos de la siguiente forma:

- 1 Entrega oportuna.
- 3 Cumple con especificaciones de entrega del trabajo incluyendo la estructura definida para el informe.
- 3 Describe adecuadamente la organización y roles asignados a los integrantes del equipo.
- 10 Realiza un análisis del enunciado y datos con el fin de diseñar una estrategia de solución al problema.
- 10 Describe las estrategias de solución al problema.
- 10 Describe los algoritmos, las funciones y las estructuras de datos utilizadas.
- 3 Expone conclusiones adecuadas al desarrollo del trabajo.
- 20 Correctitud del código cuando es ejecutado. Se realizará una prueba exhaustiva de todas las funciones solicitadas, incluyendo ingreso de datos fuera de rango, para los cuales el programa debe responder correctamente.

El informe debe estar escrito correctamente, bien diagramado y sin problemas de ortografía y redacción (se descontará 2 décimas de su nota por cada falta de ortografía en el informe).

La información contenida en el informe será evaluada de acuerdo a si esta aporta a la comprensión de la solución al problema. La solución debería ser clara y directa, por tanto, los comentarios deberían ser informativos y no extensos.

Compra y copia de tareas

Se ha detectado una red de compra y venta de las tareas semestrales, lo cual constituye una falta de ética grave no aceptada en alumnos de la Universidad de Concepción.

Es por ello que este semestre habrá una exhaustiva verificación de la autenticidad de las soluciones planteadas por los alumnos del curso. Esto significa que los alumnos integrantes de los equipos que no sean capaces de explicar los detalles técnicos de sus soluciones planteadas, tanto a nivel de algoritmos como a nivel de código, serán evaluados con nota mínima obteniendo la reprobación de la asignatura con condición NCR (No Cumple Requisitos).

Recomendación

La tarea puede ser desarrollada con cualquier intérprete usando el sistema operativo que estime conveniente, sin embargo, antes de entregar sus resultados se recomienda asegurar que estos compilen y ejecuten en JUPYTER NOTEBOOK sobre WINDOWS.

Sobre el Informe

Para confeccionar el informe de la tarea se recomienda seguir las pautas para la confección de un informe de memoria de título para las diferentes carreras en la Facultad de Ingeniería. No obstante lo anterior, a continuación se entregan algunas indicaciones para sus principales secciones:

- **Portada** El informe debe contar con una portada consistente en una página al inicio del informe con la siguiente estructura:

-
- en la parte superior derecha (encabezado) debe contener el logo de la Universidad de Concepción con el texto “Facultad de Ingeniería” y en la parte inferior de este el texto “Universidad de Concepción”.
 - en la parte superior izquierda (encabezado) debe contener el logo de la Facultad de Ingeniería.
 - en la parte central debe contener el texto “Programación” o “Lenguajes de Programación”, según corresponda, y en la parte inferior de este el texto “Tarea Semestral”.
 - en la parte inferior derecha debe contener el nombre de todos los integrantes del equipo, más abajo debe contener el nombre de la carrera de los alumnos.
 - en la parte central inferior (píe de página) debe contener la fecha con el formato “Junio de 2022”.
- **Introducción** Una o dos páginas que resuman los aspectos más importantes del trabajo.
 - **Organización** Una página en las que se identifique las funciones que asumieron como responsables los diferentes integrantes del equipo. Se solicita indicar no sólo el nombre de la función sino una breve descripción de sus principales actividades o responsabilidades.
 - **Desarrollo** Esta sección consta de varias subsecciones como por ejemplo:
 - Análisis, cuando se enfrenta al desafío de resolver un problema de mediana o alta complejidad con programación, antes de comenzar con el programa e incluso con la definición de los algoritmos, es necesario analizar cuáles son los datos, qué estrategias, qué modelos, qué procedimientos, qué fórmulas, etc. se utilizarán para pasar a la solución. Esto debe ser descrito en la sección de análisis del problema y estrategias de solución. En esta sección pueden aparecer incluso diferentes estrategias de solución las que posteriormente se pueden ir descartando para elegir la mejor o la factible.
 - Estructuras de datos, permite identificar que variables, arreglos, listas, etc. se requerirán para resolver el problema.
 - Algoritmos, permite describir los grandes bloques de funciones que se ejecutarán para resolver el problema. No es necesario describir algoritmos con niveles detallados de los procedimientos ya que esto no resultaría útil para problemas de mediana o alta complejidad.
 - **Resultados** En esta sección se puede documentar la ejecución del programa con algunos datos de entrada específicos. La descripción puede venir acompañada de algunos pantallazos correspondientes a la ejecución del programa.
 - **Conclusiones** Todo trabajo con las características del exigido en este curso requiere que una vez finalizado los ejecutores reflexionen y describan los aspectos más relevantes de su realización, respuestas preguntas como ¿cumplimos el objetivo del trabajo?, ¿qué aprendimos?, ¿que faltó aprender o resultó más complejo de realizar?, ¿qué decisiones tomamos que facilitaron la realización del trabajo?, etc. puede responderse en esta sección. No se extienda por más de una página en esto.